IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT:	Mark Moriconi, et al.
SERIAL NO.:	Unknown
S.N. of parent:	09/721,557
FILING DATE:	Herewith
TITLE:	System and Method for Maintaining Security in a Distributed Computer Network
EXAMINER:	Unknown
Examiner in parent:	W. Martin
ART UNIT:	2787
ATTY. DKT. NO:	PA1677US

CERTIFICATE OF MAILING

I hereby certify that this paper is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Box Patent Application, Commissioner for Patents, Washington, D.C. 20231, on the date printed below:

Date: January 22, 2001

David Lewis

BOX PATENT APPLICATION
COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

In response to the office action in the grandparent Application S.N. 09/248,788, now U.S. Patent 6,158,010, mailed December 30, 1999 (paper #3) please amend the specification as follows:

<u>In the specification</u>

On page 1, line 5 after “application is” insert - - a continuation of U.S. Patent Application Serial Number 09/721,557, entitled, “SYSTEM AND METHOD FOR MAINTAINING SECURITY IN A DISTRIBUTED COMPUTER NETWORK,” filed November 22, 2000, which is a continuation of 09/248,788 by the same title, filed on February 12, 1999, now U.S. Patent Number 6,158,010, which is - -.

Delete from page 8, line 3 through page 9, line 9.

On page 21, line 4 after the word “subject,” replace the word “math” with the word - - match - -.

In the claims

Cancel claims 8, 16, 23, 29, 33, 35, and 50 without prejudice or disclaimer.

Add the following new claims

57. The system of claim 1 wherein the computer readable components are associated with an application and are stored on a computer readable medium.

58. A system for maintaining security in a distributed computing environment, comprising:

a policy manager for managing a security policy; and

an application guard for managing access to a transaction related with an application as specified by the security policy.

59. A system to protect computer systems against unauthorized access for managing and enforcing complex security requirements in a distributed computer network comprising:

a policy manager located on a server for managing and distributing a policy to a client; and

an application guard located on the client, acting to grant or deny access to various components of the client, as specified by the policy.

60. The system of claim 59 wherein the server is connected via a network to the client.

61. The system of claim 59 wherein the server is connected to many clients.

62. The system of claim 59 the server further comprising:

a central processing unit;

a Read-Only Memory (ROM);

a Random-Access Memory (RAM);

a non-volatile memory;

an input device; and

a display;

wherein the ROM, the RAM, non-volatile memory, input device and display are connected via a bus.

63. The system of claim 59 wherein the policy manager is a program located on a server in non-volatile memory.

64. The system of claim 59 wherein the client contains a program stored in non-volatile memory for granting or denying access to various components or resources of the client, as specified by the policy distributed from the server.

65. The system of claim 59 wherein the server includes a non-volatile memory, where the policy manager is located that specifies the security requirements for applications and database objects;

said policy contains security rules that describe at least one constraint that constrains which applications a particular user can access and

which objects within an application a user can access.

66. The system of claim 65 wherein the policy manager allows the administrator to choose whether the constraints are effected by any of

time,

geography, and

external events.

67. The system of claim 59 wherein the policy is capable of constraining access to applications and operations within applications.

68. The system of claim 59 wherein the policy is organized into groups and hierarchies.

69. The system of claim 59 wherein the policy includes access rules, which include:

a grant rule that grants a privilege to a subject on an object under a first constraint; and

a deny rule that denies a privilege to a subject on an object under a second constraint.

70. The system of claim 59 the policy manager further comprises

a management station program to operate the policy manager,

a distributor program to distribute local client policies to clients, a logger program to track authorization requests, and

a database management system (DBMS) to maintain policy data files.

71. The system of claim 59 wherein the policy manager further comprises:

an audit log data file to record authorization requests;

an optimized policy data file;

an enterprise policy data file;

an administrative policy data file; and

a local administrative policy data file.

72. The system of claim 59 wherein the policy manager is located within non-volatile memory.

73. The system of claim 59 wherein the policy manager allows system users to

implement,

analyze,

edit and

update

a centrally-managed policy.

74. The system of claim 59 wherein the policy includes at least one policy rule comprising:

1) an object that is to be protected;

2) an access right or privilege;

3) a global or local user to which the privilege applies; and

4) conditions under which the privilege is granted or denied, wherein the user is given a choice of types of conditions including

whether to use built-in access criteria wherein the user can select

whether to use time of day and

whether to use location,

and

whether to use custom-defined access criteria.

75. The system of claim 59 wherein the policy manager comprises a Graphical User Interface (GUI) that provides at lest a user-friendly set of menu options or management services to fully operate the policy manager and control programs that perform at least

navigation,

searching,

distribution,

editing,

querying, and

log viewing.

76. The system of claim 59 wherein the policy manager comprises an Application Programming Interface (API) that at least allows programs to perform the same functions as a human operator.

77. A system comprising a computer having a security policy that includes at least one or more components having at least:

an object,

a subject,

a privilege, and

a condition.

78. The system of claim 77 wherein each object can be an application or an operation within an application.

79. The system of claim 77 wherein the object is capable of being set to be at least any of:

an application,

a method,

a web page,

a database table

a file, and

one or more menu items in a graphical user interface.

80. The system of claim 77 wherein the object can be organized into at least an object hierarchy such that:

if a user is granted a certain privilege on a parent object, then that user is automatically granted the privilege on all the children objects, and if the user is denied a certain privilege on a parent object, then that user is denied the privilege on all the children objects.

81. The system of claim 77 wherein the privilege is capable of being inherited from a parent to a child object.

82. The system of claim 77 wherein the subject is capable of being set to be at least any of:

a user and

a role containing one or more users,

who can at least

access a protected object, and

have access to at least some information in the system.

83. The system of claim 77 wherein the subject is capable of being a user that can be chosen to be either internal or external to a system.

84. The system of claim 77 wherein the object comprises a list that is capable of containing one or more users authorized to access the object who can log on to the object and be authenticated by the object through an external authentication server.

85. The system of claim 77 wherein the system is capable of having the subject be a user who at least:

can be maintained separately by one or more components each of which can be an object or directory server; and

can be extracted from said one or more components to synchronize the components thereby maintaining their access current.

86. The system of claim 77 wherein components comprise an alias-user who at least inherits all privileges of a user under certain conditions, thereby facilitating authorization management by providing fine granularity of control on propagation of a privilege.

87. The system of claim 86 wherein the system is capable of having the alias-user be created to perform a job function while the user is absent, and inheritance of the privilege takes effect only when the user is absent.

88. The system of claim 77 wherein a user of the object is capable of being defined to be local to the object.

89. The system of claim 77 wherein a user can be at least a global user mapped to a set of local users having at least one local user per object.

90. The system of claim 77 wherein the privilege defines at least one kind of access that is allowed to the object and includes at least one right to perform a particular action on the object.

91. The system of claim 77 wherein the privilege is capable of including at least

a right to execute an application,

a right to download a web page,

a right to query a database table, or

a right to view a menu item.

92. The system of claim 77 wherein the component is capable of being assigned to be a wild card that is capable of being used at least as a privilege, object, or subject.

93. The system of claim 77 further comprising an access request that includes at least:

a privilege,

an object, and

a subject;

wherein the access request is used by at least a subject to request authorization of at least a privilege on at least an object.

94. The system of claim 93 wherein the access request at least:

matches a grant rule if the privilege, object, and subject match those in the rule, and the constraint in the rule is met; and

matches a deny rule if the privilege, object, and subject match those in the rule, and the constraint in the rule is not met.

95. The system of claim 77 wherein an access request is at least:

denied if

there is a deny rule matching a request, or there are no access rules matching the request; and

granted if there are no deny rules matching the request and there is a grant rule matching the request.

96. The system of claim 77 wherein conditions comprise constraints and the system having at least facilities for defining constraints as expressions formed from operators including at least NOT, AND, and OR.

97. The system of claim 77 wherein conditions at least:

are constraints on when the object or the subject can be accessed,

specify requirements on when the access rule is applicable, and

contain options that can be set to be dependent on properties of the object or the subject.

98. The system of claim 77 wherein the system further comprises facilities for expressing constraints as at least:

1) relational operations on integers;

2) relational operations on strings; and

3) set operations.

99. The system of claim 77 wherein the system further comprises facilities that allow the user to define conditions.

100. The system of claim 77 wherein the system includes an Application Programming Interface (API) for invoking user-supplied code to evaluate user-defined functions.

101. A system comprising a computer having a security policy that includes at least one or more components having at least a set of privileges that includes at least one privilege that is capable of at least:

being granted to the user explicitly; and

being granted to a role which is granted to the user.

102. The system of claim 101 wherein:

the role is a named group of privileges containing at least one privilege that are granted to at least one user or to at least one other role; and

the at least one user granted to the role is a member of the role.

103. The system of claim 101 wherein the members of a role automatically inherit all the privileges granted or denied to the role.

104. The system of claim 101 wherein roles are organized into a role hierarchy, where parent roles are granted to children roles such that:

if a parent role is granted a privilege, then the children roles are automatically granted the privilege; and

if a role is denied a privilege, then the children roles are automatically denied the privilege.

105. The system of claim 101 wherein roles of an object may be defined as being local to that object.

106. The system of claim 101 wherein the role is at least a global role mapped to at least a set of local roles, having at least one role per object.

107. The system of claim 101 wherein the system further comprises more than one role, two of which have memberships that are mutually exclusive with respect to one another.

108. The security system comprising a policy manager located on a computer system that includes at least:

a management console or station;

a database management system;

an audit facility; and

a distributor.

109. The system of claim 108 wherein the management station further comprises a Graphical User Interface (GUI) for creating and customizing rules by system users.

110. The system of claim 108 wherein the management station supports concurrent rule development by multiple users.

111. The system of claim 108 wherein the management station includes an application guard to allow only authorized administrators to operate the management station based on at least a local administrative policy which provides a set of policy rules specifying which users are authorized to access management station.

112. A security system comprising:

at least one application guard stored on a computer readable medium and guards a protected application by preventing unauthorized transactional access to at least a portion of said application.

113. A security system comprising:

an application guard located within non-volatile memory that is designed to reside along with each protected application and supports transactional access control by allowing an application to detect an authorization service and to make authorization requests at each user interaction, data request, and business-level transaction.

114. The security system of claim 113 further comprising a distributor capable of distributing the application guard on clients throughout an enterprise.

115. The system of claim 113 wherein the application guard is integrated into the application through an application programming interface (API) or authorization library that allows the application to request authorization services as needed through an application guard interface.

116. The system of claim 113 further comprising

an authorization engine that processes an authorization request;.

a checker that parses local client policy and stores the parsed local client policy in Random Access Memory (RAM); and

an evaluator that determines whether the authorization request should be granted or denied by evaluating the authorization request with the parsed local client policy in RAM.

117. The system of claim 113 wherein the authorization engine comprises plug-ins that at least allow for additional capabilities to process and evaluate authorization requests based on customized code.

118. The system of claim 113 further comprising a logger where at least:

each authorization request is then recorded in an audit log; and

each authorization request made at a location remote from the logger is transmitted to the logger via a communication interface.

119. The system of claim 113 wherein the system is capable of implementing at least:

the application guard locally to the application; and

the application guard as a remote authorization service through a remote procedure call to another server.

120. A security system comprising:

at least one application guard stored on a computer readable non-volatile memory medium that is designed to reside along with each protected application and

guards a protected application by preventing unauthorized transactional access, and

supports transactional access control by allowing an application to detect an authorization service and to make authorization requests at each user interaction, data request, and business-level transaction;

wherein

the application guard is integrated into the application through an application programming interface (API) or authorization library that allows the application to request authorization services as needed through an application guard interface, and

the system is capable of implementing the application guard locally to the application and is capable of implementing the application guard as a remote authorization service through a remote procedure call to another server.

121. A computer readable storage medium having stored thereon a method for maintaining security in a distributed computing environment comprising the steps of:

managing a security policy via a policy manager; and

managing access via an application guard to a transaction related with an

application as specified by the security policy.

122. A method for maintaining security in a distributed computing environment, comprising:

managing a security policy via a policy manager; and

managing access via an application guard to a transaction referenced by an application as specified by the security policy.

123. A method of using a security system comprising:

using a management station to create or modify a policy rule

distributing the policy rule to appropriate clients via a communication interface included in the management station.

124. The method of claim 123 further comprises reviewing and reconstructing the policy rules via a parser to make sure that the policy rules are syntactically and semantically correct according to a predefined policy language.

125. The method of claim 123 further comprises determining via a differ-program the changes were made to optimized the policy, and wherein the step of distributing then distributes only the changed policy rules or local client policy to the appropriate application guards, which enforce access control to local applications and data.

126. The method of claim 123 wherein each application guard has its own specific local client policy.

127. The method of claim 123 further comprising optimizing via the distributor an administrative policy into an optimized administrative policy or local administrative policy for use with an application guard in the management station.

128. The method of distributing at least one security policy rule comprising:

passing the policy rule through at least

a DataBase layer (DB layer) and

an Open DataBase Connectivity layer (ODBC);

and

storing the policy rule as an enterprise policy.

129. The method of claim 128, wherein:

the DB layer formats the policy rules into standard database storage tables, and

the ODBC provides a common interface to various vendor-specific databases.

130. The method of claim 128 wherein the distribution occurs through the ODBC layer and a communication interface.

131. The method of claim 128 further comprising passing the enterprise policy to a distributor.

132. The method of claim 128 further comprising determining via an optimizer program within the distributor which application guard needs to receive the policy rules.

133. A method of configuring a security system comprising:

installing a policy manager on a server including

installing

a management station,

a distributor,

a logger, and

a DataBase Management System (DBMS);

entering a set of policy rules

installing application guards and local client policies onto client systems; and

registering plug-ins into the application guards to allow for additional capabilities in order to process authorization requests based on customized code.

134. The method of claim 133 wherein the step of entering includes presenting an administrator with the choice of whether to use a policy loader or management station to enter the policy rules.

135. The method of claim 133 wherein

if the administrator chooses to use the management station, then the step of entering includes using an edit function to enter the policy rules, and

if the administrator chooses the policy loader, then the step of entering

includes entering the policy rules into a file, and

passing the file to the policy loader.

136. A method of managing policy under management services in a management station comprising:

an authorized administrator logging into a policy manager;

the authorized administrator chooses between administrative mode to manage administrative policy or enterprise mode to manage enterprise policy;

presenting the administrator with menu options including

navigate tree,

analyze policy,

edit policy,

distribute policy,

view audit log which is a security feature that allows an administrator to view and track authorization requests have occurred at an application guard connected to a system, and

exit.

137. The method of claim 136 wherein menu option navigate tree provides a set of edit options for an administrator that include to

add,

delete, and

modify features;

wherein the administrator is presented with a choice of features on a server and on a client.

138. The method of claim 136 wherein the features that the administrator has the option apply the edit options include

global users,

global roles,

directories,

local roles,

local users,

applications,

application guards, and

declarations.

139. The method of claim 136 wherein the menu option analyze policy allows an authorized user to analyze and view rules and policies within enterprise policy.

140. The method of claim 136 wherein the user is presented with an option to

search rules, and

to query policy.

141. The method of claim 140 wherein if search rules is selected, the administrator is presented with options of searching grant rules and all the deny rules pertaining to a particular user.

142. The method of claim 140 wherein if query policy is selected, a search can be made on who is granted or denied what privilege on which objects under what conditions.

143. The method of claim 140 wherein the menu option edit policy presents an authorized user with the option to add, delete, and modify enterprise policy features.

144. The method of claim 143 wherein the features that may be edited include

a rule set,

access,

a privilege,

an objects,

an user

a role, and

an attribute.

145. The method of claim 136 wherein the menu option distribute policy includes distributing the new features of a newly entered or modified enterprise policy to appropriate application guards.

146. The method of claim 136 wherein upon selecting the distribute policy option

a distributor optimizes enterprise policy;

a differ program computes a difference between a newly optimized policy and a former optimized policy;

the newly optimized policy is then published as optimized policy in DBMS;

committing only the changed portions of the newly optimized policy to an appropriate application guard;

the application guard receives the newly optimized policy;

the application guard merges the newly optimized policy into local client policy; and

the local client policy is activated to work with the application guard.

147. A method of granting client access authorization comprising:

using an application guard that includes at least

requesting access to a software securable component associated with an application protected by an application guard, wherein the application guard constructs and issues an authorization request, and

evaluating the authorization request via the application guard according to its local client policy to determine whether to allow or deny the authorization request; and

an audit records the authorization request in an audit log;

wherein

if there is an error in the authorization request, or if the request is not valid, then the user is denied access;

if the authorization request is valid, then a determination is made whether access should be granted, and

if the evaluated authorization request does not deny access for the user, then access is allowed, and

if the evaluated authorization request denies access for the user, then access is denied.

148. The method of claim 147 wherein evaluating the authorization request includes an evaluator searching deny rules in local policy

if the evaluator finds a deny rule, then an evaluation is performed on any constraints on the deny rule

if the evaluation finds a presently valid constraint on the deny rule, then access is denied, and

if the evaluation finds that all constraints on the deny rule are not presently valid, then a search for a grant rule is performed;

and

if no deny rules are found, then a search for a grant rule is performed;

wherein after a search for a grant rule

if no grant rule is found that would allow access for the user, then access is denied, and

if a grant rule is found, then an evaluation is performed on any constraints in the grant rules wherein

if the evaluated constraint is presently valid, then access is allowed, and

if the evaluated constraint is not presently valid, then access is denied.

149. A method for securing a computer system comprising:

guarding a protected application by using an application guard to prevent unauthorized transactional access.

150. A method for providing a security system comprising:

providing at least one application guard that is storable on a computer readable medium and guards a protected application by preventing unauthorized transactional access to at least a component associated with the application.

151. A method for updating a security system comprising:

updating a set of policy rules containing at least one policy rule in a central location;

generating changes to the set of policy rules resulting from the updating step; and

distributing the changes to the set of policy rules.

152. The method of claim 151 wherein the policy rule contains entitlement information related to at least one resource.

153. The method of claim 151 wherein the policy rules are stored in a database table.

154. A method for establishing a security system comprising:

establishing a set of policy rules containing at least one policy rule in a central location; and

distributing the set of policy rules for enforcement.

155. The method of claim 154 wherein the policy rule contains entitlement information related to at least one resource.

156. The method of claim 154 wherein the policy rules are stored in a database table.

Remarks

New claims 57–156 are submitted for the Examiner's consideration and are unrelated to the prosecution history of parent application.

Non-allowed claims 1-7, 9-15, 17-22, 24-28, 30-32, 34, 36-49 and 51-56 of the parent application are being submitted for reconsideration.

35 USC §103

The Examiner made statements (the first and last sentences of the paragraph bridging pages 3 and 4, the first sentence of the full paragraph of page 6, the last full sentence on page 7, the last sentence of the paragraph bridging pages 6 and 7, the first and last sentence of the last paragraph of page 9, the first sentence of the paragraph bridging page 11 and 12, the first and second to last sentence of the full paragraph of page 14, and the middle of the last full paragraph of page 17) such as, "It would have been obvious…to *allow* Nessett to combine with Minear" (emphasis added) and as such used the wrong criterion for obviousness. The Applicants note that obvious "to allow" to combine is a standard that is easier to prove than obvious "to try" to combine because "to allow" to combine implies a lack of a sufficient motivation to actually combine or even try to combine which is present in obvious "to try" to combine. The courts, however, have stated that obvious "to try" to combine is an insufficient standard of obviousness presumably because it is only a motivation to experiment rather than to actually combine and is a easier to meet than obvious to combine (see *In re Geiger*, 815 F.2d 686, 2 USPQ2d 1276 (Fed. Cir. 1987) for an example of a case in which "obvious to try" was determined not to be obviousness). It logically follows that obvious "to allow" to combine is *a forte ore* also an insufficient criterion for obviousness under 35 USC §103.

Minear et al. relates to firewall-to-firewall connections. Minear et al.'s "invention is a system and method for regulating the flow of messages through a firewall" while in contrast Nessett et al. distributes the firewall.

The Examiner stated,

> Application gateway is known by definition as being software used to maintain security on a network.

However the phrase "application gateway" does not appear in Minear et al. Rather Minear et al. refer to an "application level gateway firewall 18." FIG. 1 labels feature number 18 as "FIREWALL." Thus the "application level gateway firewall" is just a long name for a firewall. Minear et al. and Nessett et al. discuss protecting unauthorized entry into nodes of repeaters and switches for example rather than unauthorized entry into applications and items associated with applications. The Applicants respectfully submit that the Examiner is apparently ignoring the word "application" in the phrase "application guard" of the independent claims, which implies that the securable component being guarded is in some way associated with an application.

The Examiner proposed "allowing" for a combined Nessett et al. and Minear et al. system. However, Nessett et al. states,

> Although products exist that provide for establishing security in particular product families, systems which take advantage of products in all the various categories of devices found in networks, require substantial administration. In a network involving a wide variety of network intermediate devices and terminals, an administrator is required to manage the establishment of security policy at all the various levels of protocol, and in all the various systems.

In other words Nessett et al. teach that the combined use of multiple security systems (e.g., that of Nessett et al. combined with Minear et al.) is difficult for an administrator to handle and should be avoided. Nessett et al. also states

> However, the variety of security features, and the various devices and levels of protocol at which they operate, present a significant administration problem to users of the security features. Because of the complexity, it is difficult to establish a coordinated security policy across all layers, and device types of the network, and particularly difficult to maintain such a system even if it could be successfully implemented.

Thus Nessett et al. in addition to trying to avoid using multiple security systems is trying to avoid having different security systems operating at different levels because of further administrative difficulties this creates. Minear et al.'s security system operates at the IP layer (the columns 3 lines 57-62). In contrast Nessett et al.'s system chooses the layer or layers at which to operate according to the components involved (column 3, lines 20-67). Nessett et al operates at multiple layers that are different at different parts of the system (column 4, lines 3-7) leading to a likelihood of having Minear et al. sometimes operating at a different layer than Nessett et al and sometimes operating at a common layer, thereby creating at least some places that generate the administrative difficulties Nessett et al. is trying to avoid and would be a move away from the coordinated security approach sought by Nessett et al.

Nessett et al. further teaches away from having redundancies (column 3, lines 64-67). Combining the system of Minear et al. with Nessett et al. would seem to be likely to result in undesirable redundancies at the IP layer.

Regarding claims 2, 3, 20, 39 and 40 the Examiner cited feature 11 of FIG. 1 and column 12, lines 34-37. However, the applicants respectfully submit that feature 11 of FIG. 1 and column 12, lines 34-37 do not explain any distribution mechanism. Claims 3, 20 and 40 specify a distributor for distributing the policy and further specify that the distributor is part of the policy manager. Even if Nessett et al. has a distributor it is not clear that it would be part of their policy manager, as required by claims 3, 20, and 40. Further feature 11 of FIG. 1 is a "network management station" rather than a "security management station" as specified by claims 2 and 39 from which claims 3 and 40 depend and as specified by claim 20. Column 12, lines 34-37 discuss a security policy "language" rather than a security policy (as in claim 2) or a global policy (as in claims 20 and 39) being edited via a management station as opposed to using a standard word processing editor to edit or write the source code. Possibly the systems administrator needs to edit the source code of a program that uses this language, and there is no security management station that allows for both editing and setting global or security policy. The Applicants respectfully request clarification as to how column 12, lines 34-37 imply or suggest the use of a security policy management station for both setting and editing global policy or security policy.

Regarding claims 11, and 37 the Examiner cited column 6, lines 12-34. However, column 6, lines 12-34 never mentions an application guard or authentication engine as required by claims 11 and 34 to be placed on a client server.

Regarding claims 6, 21 and 43, the "database management system" of column 9, lines 7-11, cited by the Examiner manages the "topology data base" of line 7 and is not a database management system of the claims. The topology database keeps track of the topology of the system as implied by its name. Thus the database management system (of line 8) is not for "maintaining" security policy as in claims 6, 21, and 43. Column 9, lines 23-27, cited by the Examiner, is under the heading of and therefore discusses the

"Security management back end" (column 9, line 16) rather than the database management system of line 8. The backend "translates" within the "context" of the topology database (column 9, lines 23-27). In other words the backend translates the policy using the topology database for the information about how the nodes are connected to one another rather than a database management system maintaining the security policy, as claimed.

Regarding claims 7, 28, and 46 the Examiner stated,

> Thus the security policy establishes configuration data in a repeater by updating a management node (column 12, lines 45-56).

The Applicants respectfully fail to see how the encryption of data sent between frames discussed in column 12, lines 45-56, cited by the Examiner, is related to optimizing the policy as claimed. Although encryption may enhance the impenetrability of the firewall, the policy relates to things such as whom can access which part of the system, for example, rather than the format used to send the data. Therefore encryption is <u>not</u> a way of optimizing the policy.

Regarding claims 9 and 25, the Examiner cited column 8, lines 7-14 and column 12, lines 26-31, however these lines do <u>not</u> address placing an additional application guard at each client as claimed.

It would appear that claim 19 should have been grouped with claims 9 and 25 rather than with claims 10, 26 and 36. Clarification is respectfully requested.

Regarding claims 10, 17, 26, 30, 36, and 47 the Examiner first asserts that the combination is obvious, next lists the features of the combination, then asserts that the two references are with in the same field of endeavor, and then concludes that "Accordingly" the combination would have been obvious. However, the Examiner failed to provide a motivation for the combination. Alleging that two references are within the same field of endeavor does not provide a motivation for the combination.

Regarding claim 1, the Examiner stated that "the Application level gateway serve as a guard," while regarding claims 10, 19, 26, and 36 the Examiner stated, "all network traffic must pass through one of the proxies." It is not clear why the Examiner mentions the proxies. If the Examiner was trying to allege that the proxy is the application guard, he is contradicting his earlier allegation that the gateway firewall application 18 is the application guard. If the Examiner was alleging that the proxy is the claimed interface, then it is not clear why he mentioned the encryption interface. The encryption interface is not the interface of the claims because the interface of the claims is for requesting access to securable components while the encryption interface, "Crypto interface 80 [is] used to encrypt an IPSEC payload" (column 11, lines 46-48).

It would also appear that claim 34 should not have been grouped with claims 12, 24 and 44. The Applicants respectfully fail to see how the Examiner's discussion of claims 12, 24, 34 and 44 is relevant to claim 34. Clarification is respectfully requested.

Regarding claims 17, 30 and 47 eve if *arguendo* Minear et al. teach an application guard they clearly cannot teach an application guard that guards the policy manager (as opposed to one that guards other securable components) because the policy manager the Examiner relied upon is in Nessett et al.

Regarding claims 18 and 42 it is not seen where the Examiner finds a suggestion to use a local application guard that is distributed by the policy manager. The proxies of Minear et al. are not disclosed as being distributed locally implying that they are all located in the same location.

Regarding claim 53 and 54 column 6, lines 32-34, cited by the Examiner (towards the top of page 10), do not discuss the processor and therefore do not disclose the same processor distributing the policy and executing the policy manager.

The Examiner rejected claims 5, 14, 15, 22, 32, 41, 45, and 49 under 35 USC §103(a) as unpatentable over Nessett et al. in view of Minear et al. as applied to the above claims further in view of Abraham et al.

Regarding claims 5, 22, 32, 41, and 49, the Examiner cited column 7, lines 38-43 and column 9, lines 1-10 and then stated,

> By monitoring the user transactions the policy management administrator would be able to better customize policies. This would have motivated one of ordinary skill in the art to implement the modifications set forth above.

However, the Examiner never provides support for this statement. The motivation to combine or modify a reference must also have support. The Applicants hereby respectfully call upon the Examiner to provide support for this statement in accordance with 37 CFR 1.104 d(2)or provide a reference in accordance with the last sentence of the second paragraph of MPEP 2144.03.

The Applicants respectfully note that the Examiner's discussion of claims 5, 22, 32, 41, and 49 fail to explicitly address the "audit log" of claim 32.

The Examiner stated regarding claims 14, 15, and 45,

> It would be obvious to allow the combination of Nessett and Minear to implement the system taught by Abraham. This implementation would allow Nessett distribution system used by the administrator, to be controlled by menu sets. This would offer a more efficient means for network management of the security policy system.

However, the Applicants respectfully submit that the Examiner never provides support that efficiency is in anyway recognized to be related to menus. The Applicants respectfully call for the Examiner to provide support for this statement under 37 CFR 1.104 d(2) or a reference in accordance the last sentence of the second paragraph of MPEP 2144.03.

The Examiner continued,

> One of ordinary skill in the art would have recognized that this combination would give the management abilities to analyze, edit, distribute and view audit log stored on the audit server.

However it is not clear that there is an audit log on an audit server in Abraham et al. It is not clear what would motivate the addition of an analyze option on the menu that Abraham never mentions, even assuming *arguendo* that it is present. It is not clear what support the Examiner has for the allegation that one of ordinary skill would have recognized that this combination would give management the ability to analyze and view an audit log. The Applicants respectfully call for the Examiner to provide support for these allegations under 37 CFR 1.140 d(2) or a reference in accordance the last sentence of the second paragraph of MPEP 2144.03.

Regarding claims 14, 15, and 45 the Examiner alleged that Nessett et al. teach the use of an "audit log" and cited column 13, lines 32-38. However, although Nessett et al. column 13, lines 32-38 state that

> repeaters … can monitor port disconnects and reconnects, reporting these to network management applications,

Nessett et al. fail to state that the "disconnects" and "reconnects" monitored and reported are recorded in an audit log. The words record and log are absent from Nessett et al.'s specification.

It is not clear that the options provided in column 12, lines 15-44, cited by the Examiner, are in a menu format as required in the claims. It is not clear if Abraham et al.

provides a distributed policy option. Possibly the newly edited policy only gets distributed after closing the window and/or after rebooting the system or network. As the burden of establishing a prima fascia case of obviousness is upon the Examiner, it logically follows that the burden is upon the Examiner to establish that Abraham et al. suggest an explicit menu option of distributing policy.

Regarding claim 15, Abraham et al. never discloses a menu including all of the options listed, i.e., navigate tree, analyze policy, edit policy, distribute policy, and view audit log. In particular, none of the references cited teach or suggest a menu option of analyze policy or view audit log.

The Examiner rejected claims 13 and 27 under 35 USC §103(a) as unpatentable over Nessett et al. in view of Minear et al. as applied to claim 1 further in view of Rogers et al.

In making this rejection the Examiner stated,

> Rogers does not specifically teach the loader as being al [sic] loader for bulk policies. It would have been obvious to one of ordinary skill in the art to recognize that in the event of the administrator wanting to load numerous policies the policy management system would be equipped to handle such an occurrence.

In other words the Examiner first recognized that Rogers et al. lacks a teaching for the element (the bulk policy loader) missing from the modified device of Nessett et al. and first needs to modify Rogers et al. to have this missing component. After modifying Rogers et al. the Examiner stated,

> It would have been obvious ... to allow the combination system [that of Nessett et al. combined with Minear et al.] to implement the means of a policy loader as suggested by Rogers et al.

However, the policy loader of Rogers et al. is the wrong policy loader. In other words the Examiner is assuming that obvious to include a device, that does not actually exist but is obvious over a second device, within a first device is obvious. The Applicants respectfully submit that the added level of foresight required to envision the modified second device that does not yet exist and the added foresight required to envision the behavior, potential benefits and pitfalls of this presently nonexistent device is beyond the standard of obviousness of 35 USC §103. The situation is somewhat analogous to a chess game. After any given move it can be argued that finding any and all of the next possible moves for a first player is obvious. The first player just needs to take each of his pieces one at a time and check each space along each possible direction of movement to find all possible next moves. Further after the next move has been made (no matter which move was chosen) to find all the possible moves of the next player is also likewise obvious by following the exact same process. Following this simple algorithm all possible game scenarios could be mapped out. Further, if the first player were to have all his possible game scenarios mapped out selecting which move gives the best outcome is also usually obvious. Yet, making the best next chess move is often far from obvious because of the tremendous foresight required to complete this sequence of obvious mappings and selections. Even selecting the move that will generate the best outcome were the game abruptly terminated after the next couple of moves is usually also far from obvious because of the foresight required. Likewise the foresight required to see all the possibilities after modifying a device to see the benefits and to see how to overcome potential obstacles in taking the nonexistent modified device and combining it with another nonexistent modified device is typically beyond obviousness.

The Examiner stated that combining the nonexistent modified device of Rogers et al. with the nonexistent modified device of Nessett et al. is bovious

> because Rogers teaches an invention similar to Nessett in that they both are directed towards policy management within a network system, and one of ordinary skill in the art would have recognized these similarities and concluded that they are form [sic] the same field of endeavor. This would have motivated one of ordinary skill in the art to implement the modifications set forth above.

In other words the Examiner alleged that (1) Rogers et al. and Nessett et al. are analogous art and (2) the motivation to combine these two nonexistent modified devices is that they are analogous art. However, the Applicants respectfully submit that the fact that two prior art devices are analogous art is not a sufficient motivation to combine to prove obviousness under 35 USC §103 but is just a prerequisite for using the prior art device in question.

Rogers et al.'s invention relates to administration systems (column 1, lines 10-14) and to policies for managing an entire system rather than policies that just apply one type of task such as enforcing security. Rogers et al.'s invention provides a policy implementation system that responds to changes in the network by providing threads representing event driven statements. The reason for doing this is because the written policy that would otherwise need to be implemented by operators often requires monitoring the state of the system. Rogers et al. discusses that implementation of policies often include complex rules for when to do which procedure. Rogers et al. further discuss that sequences of operations may change depending on the outcome of previous operations (column 1, lines 58-64). These problems are not very relevant to simpler policy managers such as those of Nessett et al., which just enforce access rights of workers, for example, and do not have many different types of sequences of operations that need to be changed depending on the outcome of other sequences of events. Consequently, the Applicants respectfully submit that Rogers et al. does not provide a sufficient motivation for combining his device with that of Nessett et al., with or without their respective proposed modifications.

Further although column 12 lines 15-44 of Rogers et al. cited by the Examiner may mention changing the policy, it never discusses "loading" the policy as claimed. The word "load" and variations of it never appears in the text of Rogers et al. "Loading" a policy implies importing a policy that has already been entered somewhere (e.g. in a file) as opposed to merely being entered via keyboard directly into the program. Thus the combination proposed by the Examiner fails to suggest loading the policy, as claimed.

Summary

In numerous places the Examiner appears to use an improper standard of obvious to "allow" to combine as a standard of obviousness and implies that because two references are combinable or within the same field of endeavor therefore it is obvious to combine them, which is inconsistent with established case law.

In several places the Examiner provides motivations for combining references that are not found within the prior art. The Applicants respectfully request the Examiner to either provide support for these statements under 37 CFR 1.104 d(2) or provide a reference in accordance with the last sentence of the second paragraph of MPEP 2144.03.

Nessett et al. are trying to avoid redundancies and having multiple security systems operating at multiple levels both of which are likely outcomes of combining the security systems of Nessett et al and Minear et al., mitigating against making this combination in the manner proposed by the Examiner.

The Applicants have specifically pointed out several places the Examiner alleged certain features are disclosed by the references but upon a closer look the passages cited do not appear to fully support the Examiner's assertions. For example, the Applicant respectfully submit that it is not clear which claimed elements the Examiner believes correspond to which features of the references relied upon regarding the proxy, the encryption interface and the application level gateway. The proxies of Minear et al. are not disclosed as being distributed locally implying that they are all located in the same location. The Applicants respectfully submit that a topology management system is not the same as the claimed management system for maintaining security. Abraham et al. do not disclose a menu including a log option and an analyze option. The Applicants respectfully request either a clearer explanation or withdrawal of the rejections affected.

Regarding claims 13 and 27, the proposed modification of Nesset et al. with the modified device of Rogers et al. assumes that obvious to modify a first device with a non-existent second, where the nonexistent second device is an obvious modification of a third device, is obvious. However, the Applicants respectfully submit that in the present case more foresight is required than is justified under obviousness of 35 USC §103 for such a compound modification.

Rogers et al.'s concerns regarding changes in sequences of operations do not seem very relevant to the simpler system of Nesset et al. and therefore do not provide a sufficient reason for making the compound modification proposed by the Examiner.

In conclusion, Applicants respectfully submit that the claims are allowable, and therefore request that the Examiner withdraw the rejections and pass the application to issue. If the Examiner has questions regarding this case he is invited to contact Applicants' undersigned agent.

Respectfully submitted,

Richard Cappels et al.

Date: January 22, 2001 By: David Lewis

David Lewis, Reg. No. 33,101
Carr & Ferrell, *LLP*
2225 East Bayshore Road, Suite 200
Palo Alto, CA 94303
Phone: (650) 812-3400
FAX: (650) 812-3444